COMPUT NEWS

University of Toronto

December 78 ISSN 0315-4661 No. 154

33632

COMPUTERNEWS is a monthly publication produced by the University of Toronto Computer Centre. Enquiries or contributions should be directed to the Editor, Janet Campbell or the Assistant Editor, Cathy Gillevet in the Engineering Annex, room 206 or (416)-978-2066. Layout and ATS typing are provided by Kathryn Bourne. Subscription changes should be directed to the Information Office, Engineering Annex, room 206 or (416)-978-4990.

……………………………………

Permission is granted to reprint articles from COMPUTERNEWS for noncommercial purposes provided the author, publication and issue are acknowledged.

University of Toronto
Computer Centre
11 King´s College Road
TORONTO Ontario
M5S 1A1

416/978-2066

Janet Campbell, Editor

TABLE OF CONTENTS

PAGE

The DEC 1060 1
DEC News 1
Time-sharing and DEC 1060 Advice 3
Opening DEC 10 Accounts 3
DEC Disk Storage Quotas 4
DEC 10 Education Schedule 5
DECsystem-10 Manuals 6
What´s New with Terminals 8
BASIC 9
SOS: A Text Editor on the DEC 10 12
P.S. ... Programming Language-1 13
Changes in Assembler Support 14
GPSSV Now Available 14
WYLBUR Words 15
Recent Acquisitions in the Department of Computer Science Library 16
UTCC Systems 17
UTCC Directory 18

Digitized by the Internet Archive
in 2018 with funding from
University of Toronto

https://archive.org/details/computernewsdece154jane

THE DEC 1060

The University of Toronto Computer Centre is pleased to announce that the new DEC 1060 general purpose time-sharing computer system will soon be available for use. The DEC 1060 is scheduled to commence operation by the end of January, 1978.

Time-sharing takes maximum advantage of the capabilities of the computing system in that it allows many independent users to share the facilities of the DECsystem-10 simultaneously. Because of the interactive, conversational, rapid response nature of the time-sharing system, a wide range of tasks can be performed by many users concurrently. The facilities of the TOPS-10 (Total Operating System) are available to time-sharing users at remote terminals. The user interacts with the system while the program is executing. Thus, interactive computing gives the impression that the individual user has sole access to the computer's facilities.

This highly developed interactive system is but one of the many benefits offered by the DEC 10. Ten years of software development have produced an operating system which places a powerful command language, a full range of software and a flexible, easy to use file system at the finger tips of the programmer. With the DEC 1060, the use of JCL is not required to execute a job. Several of the language processors available on the DEC 1060 can be used with the appropriate version of Dynamic Debugging Technique (DDT). The ability to interactively debug programs virtually eliminates the need for printed program dumps.

The DEC 1060 will offer versions of such standard language processors as FORTRAN, PL/1, COBOL and ASSEMBLER. In addition, the processors BASIC, SOS and TECO will be available. BASIC is a programming language primarily designed for use in an interactive environment. This language is both easy to learn and simple to use. TECO and SOS are two powerful editors which are available for text preparation. TECO, a character oriented editor, is reputed to be the world's most powerful text editor. SOS is the most popular line-oriented editor.

UTCC believes that the DEC 1060 will provide users with a valuable computing tool. For information concerning the present status of the DEC 1060 System and the expected rate schedule for DEC users, refer to the article in this issue of COMPUTERNEWS entitled, DEC NEWS.

Stan Yagi

DEC NEWS

DEC 1060 Status Report

The basic DEC 1060 hardware has been installed and is now operational. However, several testing and installation procedures must be completed before the DEC 1060 will be available to users.

cont'd...

DEC NEWS cont'd

With respect to testing the equipment, the DEC technical staff are in the process of running the final checkout tests. Although the TOPS-10 operating system has been installed, several of the language processors offered by the DEC 10 are currently not supported. As well, some of the telecommunications hardware is yet to be installed. Another factor influencing the final implementation date is that UTCC staff are in the process of developing the IBM S/370-165-II link software to allow high volume printing and RJE to the 370/165.

Once these procedures have been completed, the DEC 1060 will commence full operation. The final implementation date will be announced through the standard UTCC channels.

Proposed Rate Schedule

The basic rate schedule which UTCC will recommend to the Advisory Committee on Computing Facilities and Services (ACCFS) is outlined in the following.

Rate Schedule Prior to September, 1978

From the time the DEC 1060 commences operation until the beginning of September, the regular charges for interactive computing will be substantially reduced. During the first month of operation, UTCC will be offering a 100% discount on computing charges. This month of free computing will enable users to experiment without incurring charges. Once the initial month of operation has expired, a discount of 50% will be offered until the beginning of September.

Rate Schedule As of September, 1978

The proposed rate schedule, which will be effective September, 1978, is as follows.

Connect time	$3.00 per connect hour
CPU time	$2.10 per CPU minute
Memory	$0.18 per Kiloword-minute
Disk I/O	$0.60 per 1000 disk I/O
Storage	$0.001/block per day

In addition to the discounts being offered during the initial period of operation, an additional discount will be available to users. This discount applies to jobs which are run during non-prime time. Users of the S/370-165-II should be familiar with this discount. For users who are not aware of the discount, it is related to the service factor. The service factor is used to calculate the overall cost to the user by multiplying all costs incurred by the appropriate factor. The service factor is 1.00 during prime time and .60 during non-prime time. The following table outlines the hours of prime time and non-prime time service.

		Service Factor
Prime time	Weekdays 8:00 to 19:00	1.00
Non-Prime time	Weekdays 19:00 to 8:00	.60
	Weekends 0:00 to 24:00	.60

It should be noted that non-prime time discounts will not apply to jobs printed on high speed printers or to online storage.

cont'd...

DEC NEWS cont'd

The proposed rate schedule is based on the estimated usage and system load. When the DEC 1060 commences operation, the system load may not be in accordance with UTCC's estimation. If this is the case, the rate schedule will be adjusted to reflect the difference. The final rate schedule will be made available via USERBOOK.

Questions concerning the DEC 1060 rate schedule should be directed to Stan Yagi, a Customer Service representative, or a Computing Co-ordinator.

Stan Yagi

TIME-SHARING AND DEC 1060 ADVICE

UTCC is reorganizing and supplementing its advising services in order to provide our time-sharing users with an advising service which is comprehensive and easily accessible. Our investigations have shown that our phone-in advising services are becoming increasingly popular with our user community and accordingly, we are expanding these services to meet the needs of our time-sharing users. The phone-in advising services are outlined below:

DATATALK

DATATALK will continue to provide advising and support services for SHARP APL. In addition, users should contact DATATALK for advice on DEC APL and BASIC. DATATALK services are available by calling 978-6710, 9AM-6PM, Monday through Friday.

TIME-SHARING ADVICE

A new phone-in advising service specializing in time-sharing problems is being set up. The new service will provide advice on DEC problems (with the exception of those covered by DATATALK) and on all other UTCC time-sharing services. A new staff member will be added to the advising staff of the Engineering Annex Terminal to assist with the additional work load.

The new service will go in to effect when the DEC 1060 becomes operational and the phone number and hours of service will be announced at that time.

In addition to the phone-in advising services, time-sharing users are reminded that they may obtain help from any UTCC Computing Co-ordinator.

Don Gibson

OPENING DEC 10 ACCOUNTS

In order to use the DEC 1060, a user will need a Service Access Code (SAC) to identify himself within the computer. Because the DEC 10 is significantly different from any other system available at UTCC, a user will require a new number.

A DEC 10 SAC is comprised of two parts, a Project Number and a Programmer Number. This combination is usually referred to as a Project-Programmer Number (PPN). Note that, for security purposes, a PPN should never be used by more than one user.

cont'd...

OPENING DEC 10 ACCOUNTS cont'd

Some of the rules regarding the assignment of each of these numbers are outlined below.

A given Project Number cannot be assigned to more than one project. However, several users can have the same Project Number as part of their PPN if they are all working on the same project. For example, a course instructor may arrange to have the same project number assigned to all the students taking his course. Users working together who wish to share files (both programs and data) can facilitate this by obtaining PPNs with the same Project Number.

Similarily, a given Programmer Number should not be assigned to more than one user. However, an individual user who is working on more than one project can reference his files from any project by obtaining PPNs with the same Programmer Number.

In order to obtain DEC 10 SACs, a user need only specify his Customer Account Number (CAN) and the number of PPNs he requires. However, other information can be specified; this includes:

1. multiple Project Numbers (default-unique)
2. multiple Programmer Numbers (default-distinct)
3. disk storage quotas (defaults-200 block logged-out, 500 blocks logged-in. For details see DEC DISK STORAGE QUOTAS in this issue of COMPUTERNEWS)
4. dollar limit (default-no limit)
5. expiry date (default-no expiry)
6. name of user of PPN (default-CAN holder)

This information should be supplied to Gail Cavers, Access Code Clerk (Engineering Annex, Room 206, 978-8703) or to the local Computing Co-ordinator.

Henri Schueler

DEC DISK STORAGE QUOTAS

Storage on the DEC 10 is based on storage blocks, where one storage block is 128 36-bit words or approximately 640 characters. A rough rule of thumb is that 20 blocks is equivalent to one IBM 3330 track. The storage quotas are tied to the user's Project Programmer Number (PPN) and can be set at two levels. The "logged-out" quota represents the amount of storage that a PPN can own when the user is logged off of the computer. Conversely, there is also a "logged-in" quota which could be set higher than the logged-out quota, allowing the user a form of temporary storage. This space would have to be released before the user could log off.

Bill Foster

DEC 10 EDUCATION SCHEDULE

A series of short courses is currently being planned for early 1978 to introduce users to UTCC's new DECsystem-10. Since these courses will involve giving attendees "hand-on" experience at typewriter terminals, class size will be determined by the number of terminals that can be provided at a given time. Those who would like to attend any of the following courses should contact the Information Office, Room 206, Engineering Office, 978-8995 as early as possible. Please keep in mind that a UTCC Computer Account Number (CAN) is required in order to register for any course offered by the Centre.

1. Introduction to Programming Using BASIC

Prerequisite: A knowledge of computer concepts would be helpful.

BASIC is a conversational, problem solving language that is well suited for use in a time-sharing environment. BASIC is easy to learn and use. BASIC has been widely applied in the scientific, business and educational communities. As well, BASIC can be used to solve both simple and complex mathematical problems from the user's terminal.

This course will be aimed exclusively at the non-programmer. While learning the concepts of interactive computing and communicating with the DEC 10 operating system, the student will be taught how to code simple applications in the BASIC language. The objective of the course is to give the student a foundation on which to build his/her own programming knowledge and expertise.

2. Introduction to DEC 10 for "Card" Programmers

Prerequisite: Experience in writing and running programs is required.

Oriented towards those currently using the batch processing mode of operation, this course will introduce the concepts of interactive computing on the DEC system-10. On completion, the student should be able to communicate with the TOPS 10 operating system using the basic commands of the control language and should be familiar with the available user service programs or "Utilities".

3. Introduction to DEC 10 for "Terminal" Programmers

Prerequisite: Same as #2.

This course is the same as #2 except that a knowledge of interactive computing will be assumed. The approach will be to discuss and illustrate the essential differences between conversational programming on the DEC 10 and TSO, WYLBUR, etc.

4. Introduction To Text Editing Using SOS

Prerequisite: No previous exposure to computing is required.

cont'd...

DEC 10 EDUCATION cont'd

For those involved in the preparation of printed documents, typing, editing and retyping can be a tedious and time-consuming business. In cases where an extensive document must undergo frequent revision, the procedure becomes even more cumbersome. SOS, a fast and easy-to-use text editing system, allows the typist to make corrections and revisions without having to manually reproduce entire pages or sections. Of the many text editors available at the Centre, SOS is considered to be the most advantageous from the point of view of ease of learning, searching and editing facilities and confidentiality protection. RUNOFF, a utility used to format final copy, will also be introduced.

5. Introduction to SOS for ATS Users

Prerequisite: Experience using ATS or a similar computerized editing system.

This course is the same as #4 except that the approach will be a discussion of the conceptual differences between SOS and ATS, NSCRIPT, WYLBUR, TSO, etc.

Marg Doherty

DECsystem-10 MANUALS

The UTCC Information Office is currently in the process of making available to users a complete reference set of documentation for both DEC 10 hardware and software. Users wishing to obtain their own copies of certain manuals may arrange to receive them through the Information Office either by placing a special order or, in the case of commonly used publications, picking them up as they become stock items. The following is a list of those manuals for which we expect the demand to be greatest. Many of these are now available in limited quantities at the Information Office, Room 206, Engineering Annex, 978-4990.

Getting Started With DECsystem-10, DEC-10-XGSDA

This manual introduces the user to the DECsystem-10 and the TOPS-10 Operating System. Intended to be used while siting at a terminal, it includes instructions on logging into the system, entering programs, executing programs, editing programs, working with files and using BASIC.

Getting Started With TOPS-10 Commands, DEC-10-OTSCA

This manual is a simplified guide for the time-sharing user. The commands described in this document are the subset of monitor commands that are most useful to the novice user. With these commands, the user can create, edit, and manipulate files; translate, execute, save and start programs; use devices; and obtain information from the system.

Utilities Manual, DEC-10-UTILA

This manual offers instructions on the use of various utility or service programs: CREF, DDT, FILCOM, FILEX, GLOB, OPSER, PIP and RUNOFF.

cont'd...

DECsystem-10 MANUALS cont'd

These programs are used to obtain cross-referenced listings of program codes, check out and list on-line programs, compare files and output differences, convert core image formats, facilitate multiple job control, transfer files from one device to another, and format documents.

Sort/Merge Users' Guide, AA 0997C

This document is written for users who are sorting data into a specific sequence and/or merge files. Although the reader need not have programming skills, he should be familiar with computer operations and with data types and recording modes.

Introduction to TECO, DEC-10-UTECA

TECO is a character-oriented editor designed to modify source documents written in languages such as FORTRAN, COBOL, and MACRO-10. The manual is written as a tutorial for those who have never used TECO.

TECO, DEC-10-UTPRA

This manual is a complete reference guide for the advanced TECO user. It is not designed to be used as a beginner's text, and those learning TECO should not use it as such.

APLSF Programmers Reference, DEC-10-LPLSA

This manual presents two implementations of the APL language: APL and APL-SF. APL is a subset of APL-SF. The major difference between the two implementations is that APL-SF has a file system. Both versions support shared system functions and system variables. This manual is not intended to be used as a language primer, but rather as a reference manual for all users of APL and APL-SF. The manual will be useful as an introductory text for novice APL users having computer experience and some background in mathematics.

BASIC Conversational Language, DEC-10-LBLMA

This manual describes, in a somewhat tutorial style, features of the problem-solving language called BASIC. This manual describes elementary procedures, such as how to gain access to BASIC, as well as sophisticated techniques, such as RESTORE, CHAIN and PAGE statements.

CPL Users Manual, DEC-10-LCPLA

CPL (Conversational Programming Language) is an interpreter supporting a subset of the ANS PL/1 language. Non-programmers and novice programmers find this language easy to use. This publication is intended to be both a tutorial and a reference manual.

COBOL-68 Programmers Reference, DEC-10-LCPRA

This manual describes COBOL as it is implemented on the DECsystem-10. It assumes that the reader has a knowledge of the COBOL language and is intended to be used primarily for reference purposes. It is not a tutorial guide for beginner COBOL programmers.

cont'd...

DECsystem-10 MANUALS cont'd

FORTRAN-10 Programmers Reference, AA-0944E

Written at the level of a reference manual, this document assumes that the reader has a rudimentary knowledge of the FORTRAN language. It is a guide to implementing FORTRAN on the DECsystem-10.

MACRO-10 Assembler Programmers Reference, DEC-10-LMCOA

This document was written for the experienced Assembler programmer. It describes how numbers are handled, arithmetic and logical operations, syntax of MACRO statements, writing macros, and interactive use of MACRO and Assembler output.

LINK-10 Programmer's Reference, DEC-10-ULKMA

This reference guide is aimed at the intermediate-level applications programmer. It contains complete documentation on LINK-10 including descriptions of the LINK Item Types generated by the DECsystem-10 Translators.

SOS User's Manual, UM-5B

SOS (Son of Stopgap) is a line-oriented text editor. It is powerful, easy to use, and contains features not available using other DECsystem-10 editors. Some of the features offered by the SOS editor are extended searching capabilities, text justification, and file encryption. This manual is written as a reference document and, as such, those without a sufficient knowledge of computing concepts and terminology will find it difficult to follow when attempting to learn SOS.

Marg Doherty

WHAT'S NEW WITH TERMINALS

Additional low speed terminals have been purchased and will be placed in public terminal areas, to provide access to the DECsystem-10. Both CRT terminals (Volker Craig VC303A) and hardcopy terminals with APL (DEC writer II LA36) have been purchased for the terminal areas.

Accommodations have been found for most of the terminals; however, UTCC has run into difficulties acquiring space in the Burton Tower and the Sidney Smith Building. UTCC has been told that there is a shortage of space in the Burton Tower. The Physics, Computer Science and Astronomy departments will need terminals to access the DECsystem-10 and UTCC is hopeful that space will be found to house five terminals in the Burton Tower. UTCC is still trying to find spaces to set up a terminal area in the Sidney Smith Building.

The following list summarizes the types and locations of terminals available to DEC 1060 users at UTCC:

LOCATION	TYPE	QUANTITY
Burton Tower	LA 36	2
See Co-ordinator	VC303A	3
Room 1202		
Eng. Annex	LA 36	6
Terminal Room	VC303A	10
Erindale College	LA 36	8
Room 2045	VC303A	5
Scar. College	LA 36	4
Room S-624	VC303A	4
Sidney Smith	LA 36	1
See Co-ordinator	VC303A	6
Room 2100		

cont'd...

WHAT´S NEW cont´d

Anyone in the University community interested in purchasing (at reduced rates) or leasing a VC303A terminal, a LA 36 terminal or a high quality print terminal should contact Kurtis Bishop at 978-6133 for additional information.

Ralph Lombardi

BASIC

BASIC is a programming language which is intended to be used primarily in an interactive programming environment. The language, which displays many similarities to FORTRAN, is simple to learn and use. It is widely available on many machines and operating systems; in fact, it is the only high level language available on most micro-computers. Its wide availability has also meant that it is the language used in many program packages.

Until the acquisition of UTCC´s DEC 1060, the language processor BASIC has not been generally available to the UTCC community. However, the DEC 10 offers a powerful version of BASIC. It is expected that BASIC will rapidly become the principal language processor in use on the DEC 10 machine. BASIC´s popularity at other installations and its general availablility attest to its simplicity and flexability.

The BASIC language contains its own editor (although other, more powerful, editors can be used). Since the code is interpreted rather than compiled in the regular sense, the system is capable of giving clear error messages. In addition, the combination of interpreted code and a built-in editor allows the system to rapidly switch from program execution to program development and correction -- a feature which allows for rapid program development.

The best way to describe the nature of the BASIC language is with a simple example. Suppose that a user wishes to develop a program to calculate the average of up to 100 numbers. The terminal session that might arise to solve this problem is shown in figure 1. (Please note: in figure 1 the characters typed by the DEC 10 computer are shown in upper case and those typed by the user are in lower case).

A user at the terminal would begin the BASIC session by logging into the DEC 10 time-sharing system and then, in response to the monitor prompt (a period), by typing the command:

```
.r basic
```

To this the machine would reply:

```
READY, FOR HELP TYPE HELP
```

This reply means that the BASIC system is now ready to receive commands from the user. At any point (except during the time that the system is executing a user program and waiting for input) the user may type "help" to obtain a list of standard BASIC commands with their meanings.

cont´d...

BASIC cont'd

The user must first tell the system that he wishes to enter a new program. At the same time, he will also assign a name to the program. This is done by entering the command:

```
new avg
```

This means, "I wish to create a new program, called 'AVG'".

The user is now ready to run his program on the DEC 10. In the BASIC language, each line of a program begins with a line number. The user tells the system that he is typing in a line of his program by beginning that type-in with a line. The user can enter the program lines in any order he pleases, but the machine will retain them in numerical order. For example, the user might type the following to enter his program:

```
10  dim v(100)
20  print "give numbers"
30  mat input v
40  let n=num
50  if n=0 then 999
60  for i=1 to n
70   let s=s+v(i)
80   next i
90  print "the average is";s/n
100 go to 20
999 end
```

The first statement (number "10") defines "v" to be a vector containing up to 100 numbers. The second line will cause the computer to print a message at the terminal reminding the program user of what information he must enter. The third line will allow the program user to enter these numbers from his terminal (the exact format of his response will be shown below). The fourth line uses the built-in function "NUM" which can provide the program with the number of numbers read in by the previous statement; this is assigned to variable "N". Line "50" tells the machine to branch to line "999" (the end of the program) if N equals 0; i.e. if no numbers were entered.

At this point, the user realizes that he has left a line out. Lines "60" to "80" calculate the sum of the numbers that have been read in (the FOR/NEXT construct is similar to a FORTRAN DO-loop), but the user has forgotten to enter the line which initializes the variable containing the sum to zero. Therefore he inserts a line (number "55") by typing:

```
55  let s=0
```

Lines "90" and "100" print the average and cause the machine to go back to line "20" so that a new average may be calculated.

The user now wishes to see exactly what his program looks like. The "LIST" command causes the machine to type back the program listing shown in figure 1. The program looks correct now, so the user tries it out by typing the command "RUN".

The execution of the user's program is also shown in figure 1. Note that the user can enter a series of numbers on one line in reply to the input requested by the "MAT INPUT" instruction. If he wishes to continue onto a second line, he ends his input with an ampersand ("&").

cont'd...

BASIC cont´d

This tells the machine that more data is to follow. When he tires of calculating averages, he merely enters no numbers when input is requested. His program will then cause the machine to branch to the end of the program thereby terminating execution.

Now that the program works properly, the user wishes to clean up the statement numbers that he is using. He does this with the "RESEQUENCE" or "RES" command. When he lists the program again after executing the "RES" command, he will note that the statement numbers have been reassigned, and furthermore, that statement number references within the program statements themselves have also been updated.

To save the program so that it can be accessed during a later BASIC session, the user types the "SAVE" command. Then, to end his BASIC session, the user types the command "BYE".

Figure 2 shows a later BASIC session. Here the user recalls the "AVG" program (using the "OLD" command), lists the program and changes line 100 to print both the sum and the average of the numbers that he has entered. After testing this new version, he replaces the old version with this new one and ends his session using the BYE command.

The above example has hopefully given the reader some idea of what the BASIC system on the DEC 10 is all about. However, an understanding of this example does not make the reader a BASIC expert since many important procedures and features have not been discussed. The following DEC 10 manuals are recommended to assist the reader in becoming familiar with BASIC on the DEC 10:

"Getting Started With DECsystem-10"
-- provides a basic introduction to the time-sharing system commands and to the BASIC system.
"BASIC Conversational Language Manual"
-- provides a detailed, tutorial-like introduction to the BASIC system.

In addition, questions regarding BASIC on the DEC 10 may be directed to John Bradley (phone 978-3995) or Datatalk (978-6710).

John Bradley

FIGURE 1 -- A BASIC Terminal Session:

```
    .
    .
 .r basic
 READY, FOR HELP TYPE HELP
 new avg

READY
10   dim v (100)
20   print "give numbers"
30   mat input v
40   let n=num
50   if n=0 then 999
60   for i=1 to n
70    let s=s+v(i)
80    next i
90   print "the average is";s/n
100  go to 20
999  end

55   let s=0
list

AVG      16:13      03-NOV-77
10   DIM V(100)
20   PRINT "PRINT NUMBERS"
30   MAT INPUT V
40   LET N=NUM
50   IF N=0 THEN 999
55   LET S=0
60   FOR I=1 TO N
70    LET S=S+V(I)
80    NEXT I
90   PRINT "THE AVERAGE IS";S/N
100  GO TO 20
999  END
```

cont´d...

BASIC cont´d

```
READY
run

AVG       16:16       03-NOV-77

GIVE NUMBERS
 ? 1,5,4,7,3,6,4,1.5,1.5
THE AVERAGE IS 3.66667
GIVE NUMBERS
 ? 1,4,6,3,2,6,5,4,3,8,2,1,&
 ? 5,7,4,3,2,6,2,6,9,0,1,6,3
THE AVERAGE IS 3.96
GIVE NUMBERS
 ?

TIME:  0.15 SECS.

READY
res
list

AVG       16:19       03-NOV-77

10   DIM V(100)
20   PRINT "GIVE NUMBERS"
30   MAT INPUT V
40   LET N=NUM
50   IF N=0 THEN 120
60   LET S=0
70   FOR I=1 TO N
80    LET S=S+V(I)
90    NEXT I
100  PRINT "THE AVERAGE IS";S/N
110  GO TO 20
120  END

READY
save

READY
bye
    .
    .
    .
```

FIGURE 2 -- reusing a previous BASIC program

```
    .
    .
 .r basic
 READY, FOR HELP TYPE HELP
 old avg
 list

AVG       12:10       05-NOV-77

10   DIM V(100)
20   PRINT  "GIVE NUMBERS"
30   MAT INPUT V
40   LET N=NUM
50   IF N=0 THEN 120
60   LET S=0
70   FOR I=1 TO N
80    LET S=S+V(I)
90    NEXT I
100  PRINT "THE AVERAGE IS";S/N
110  GO TO 20
120  END

READY
100  print "sum:";s;",average:";s/n
run

AVG       12:11       05-NOV-77

GIVE NUMBERS
 ? 5.5, 4.5, 8, -2
SUM: 16 ,AVERAGE: 4
GIVE NUMBERS
 ?

TIME: 0.10 SECS.

READY
replace
bye
    .
    .
    .
```

SOS: A TEXT EDITOR ON THE DEC 10

There are several text editors available on the DEC 10 system. They include TECO, LINED and SOS. All feature powerful text editing facilities; however, it is the SOS text editor that combines the best features of the other two editors. The version of SOS available at UTCC is the latest version offered by the University of Arizona.

cont´d...

SOS: A TEXT EDITOR cont´d

One of the prime advantages of SOS is that it is easy to learn and easy to use. It is used to input and edit files. Its features include insertion, deletion, replacement, searching and substitution of text. As a result, lines which require minor corrections do not have to be completely retyped. The search command allows for "context editing" in the sense that the user can instruct SOS to find a particular string (i.e. a word, part of a sentence, etc.) instead of manually searching through an entire listing of the file.

SOS features copy and transfer commands which allow the user to move text around in a file. The copy command can also copy lines from one file to another file. Other features include text justification and file encryption; the latter provides for data security. SOS also allows a file to be subdivided into pages.

SOS users have access to a HELP file. Detailed information on a specific SOS command can be printed out at the user´s terminal by typing the word HELP followed by a blank and the command.

As mentioned earlier, SOS is used to input and edit a file. In order to format a file, SOS can be used in conjunction with a text formatter, called RUNOFF. RUNOFF´s commands include text justification, underlining, centering text, page headings/subheadings, page numbering, and footnoting.

SOS is documented in the SOS Users Manual; detailed instructions on the RUNOFF text formatter are contained in the DEC Utilities Manual. Questions regarding the SOS text editor and the RUNOFF text formatter should be directed to either Ian Darwin at 978-7318 or Ruth Borenstein at 978-6355.

Ruth Borenstein

P.S. ... P.S. ... P.S. ...

PROGRAMMING LANGUAGE-1

FORTRAN, COBOL, and ALGOL were the principal forerunners of the programming language, PL/1 (Programming Language-1). FORTRAN became the language used primarily by engineers and scientists in solving numerical problems. COBOL is widely used in business and industry for solving large data-handling problems (one such application of COBOL is payroll management). ALGOL, which was designed by a team of international computing experts, is similar to FORTRAN in that it is primarily designed for solving numerical problems.

PL/1 came into existence in an interesting manner. The following is a brief account of how PL/1 originated. Large-scale IBM users convened to form the SHARE organization. The intention of this organization was to augment the efficiency of computer systems through the sharing of programs and information. In October, 1963, an Advanced Language Development Committee was formed within SHARE to develop a general, multi-purpose programming language.

cont´d...

P.S. ... PROGRAMMING cont´d

It was hoped that this new language would resolve many of the problems that were arising with the use of FORTRAN, COBOL, or ALGOL. The aim of the SHARE Committee was to design a programming language which would serve both science and business oriented computer users. As well, the language was to provide an easy-to-learn structure which would be usable by the novice programmer as well as by the experienced programmer. It was intended that this language could be supported on a wide variety of computer systems.

In March, 1967, a report defining the first version of NPL (New Programming Language) was issued by the SHARE Committee. During the subsequent revisions which were made to the language, the name PL/1 was decided upon.

Today, PL/1 is a rich and flexible programming language. Because of the many advantages offered by this language, PL/1 is the logical choice in Programming Services´ applications.

Isabel Dunlop

CHANGES IN ASSEMBLER SUPPORT

Editor´s Note: This article is reprinted with changes at the author´s request.

UTCC is updating its support for the Assembler language. Changes are being made to the JCL procedures and ASSEMBLER. The old ASSEMBLER procedures were deleted on January 9, 1978 and the procedures documented in USERBOOK are the only ones available. The old and current procedures are summarized below:

Old Procedures	Current Procedures
ASMFLDGO	ASMGO
ASMFORTG	ASM, then FORTGO
ASMGC	ASM,LEVEL=G
ASMGCG	ASMGO,LEVEL=G
ASMGCL	ASMLE,LEVEL=G
ASMGCLG	ASMLEGO,LEVEL=G

UTCC is installing the latest release of ASSEMBLER G and the parameter string is not fully compatible. Differences will be detailed in an update to USERBOOK.

Jim Heifetz

GPSSV NOW AVAILABLE

IBM´s General Purpose Simulation System (GPSS) is perhaps the most widely used simulation package at UTCC. For several years now, UTCC has offered a version of GPSS known as GPSS/360. Although this version is outdated and is no longer supported by IBM, for the most part it functions satisfactorily. Some bugs do exist but do not influence the simulation results (e.g., the Output Editor does not edit properly).

UTCC has recently installed the latest version of GPSS called GPSSV.

cont´d...

GPSSV NOW AVAILABLE cont'd

A few of the new features available in GPSSV are: interface capability with user-written FORTRAN and PL/1 programs, a maximum of 1020 parameters for each transaction, "unavailable" and "available" attributes for equipment entities, a cross reference listing of all symbolically addressed entities, error messages that include the reason for the error as well as the error number, and free-form statement coding. Most models developed and run under GPSS/360 can be run under GPSSV without modification. A good discussion of the additional features available in GPSSV can be found in Appendix A of "Simulation using GPSS" by T.J. Schriber (published by J. Wiley and Sons).

The necessary JCL to run a GPSSV job is as follows:

```
// EXEC GPSSV
//GPSS.SYSIN DD *
     .
    GPSSV program and data
     .
/*
//
```

At present there are no plans to discard the GPSS/360 simulation package. Initial tests indicate that both GPSS/360 and GPSSV function properly under MVS.

GPSSV manuals can be obtained through the Information Office in Room 206 of the Engineering Annex. Any questions or problems should be directed to Bob Chambers in Room 1202 of the Burton Tower (phone: 978-8823).

Bob Chambers

WYLBUR WORDS

ROUTE and PRIORITY cards

Many WYLBUR users still insert /*ROUTE and /*PRIORITY statements in their JCL files, and modify these statements before RUNning the jobs. A more convenient and less expensive method is to use the DEST and PRIORITY options on the RUN command. For example:

```
?run pri 4 dest phys
```

will submit a /*PRIORITY 4 and a /*ROUTE PRINT PHYS card along with the JCL in the file.

Subsystem Died?

WYLBUR users may occasionally receive the message:

```
YOUR SUBSYSTEM HAS DIED.
YOU WILL BE RETURNED TO MILTEN
```

This indicates that WYLBUR has failed ("abended"), but the operating system and MILTEN (WYLBUR's terminal driver) are still active and the user is connected to MILTEN. Most commands are invalid at this time. In this instance, the user should wait a minute or two, then give the response "WYLBUR" to the "SYSTEM?" prompt. If the response obtained is, "WYLBUR: NOT AN ACTIVE SYSTEM", again the user should wait a minute; by this time the operator should have restarted WYLBUR.

Ian Darwin

RECENT ACQUISITIONS IN THE DEPARTMENT OF COMPUTER SCIENCE LIBRARY

Aho, Alfred V. and Ullman, J.D.
Principles of compiler design.
Reading, Mass., Addison-Wesley, 1977.

APL Congress.
APL 76: Proceedings, ed. by G.
Truman Hunter.
New York, ACM, 1976.

Association for Computing Machinery.
Proceedings of the national
conference, 1976.

Stark, Richard H. and Dearholt, D.W.
Computer concepts and assembler
programming: 360/370 systems.
New York, Academic Press, 1975.

Tuggle, Sharon K.
Assembler language programming:
systems/360 and 370.
Chicago, Science Research Assoc., 1975.

UTCC SYSTEMS

SYSTEM/370 MODEL 165-II

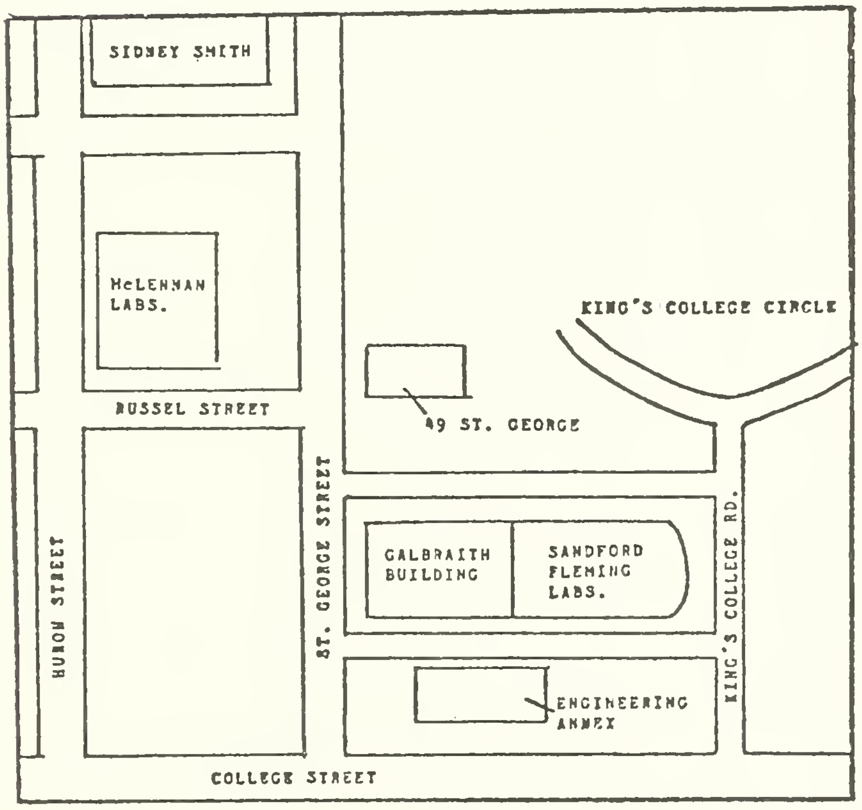
- located in SF103/105/112
- provides General Purpose Job Stream, High Speed Job Stream, TSO, WYLBUR
- 4 megabytes of memory
- OS/MVT with HASP

SYSTEM/360 MODEL 65

- located in McLennan Physical Laboratories
- provides APL and ATS services
- 1536K bytes of core
- OS/MVT with APL and ATS

COMPUTER RESEARCH FACILITY

- located in SF207
- DEC GT44 System with PDP-11/40 CPU
- DEC GT40 System with PDP-11/05CPU
- provides specialized graphics and interactive graphics
- provides on-line and real-time computing services, data acquisition and minicomputer services



UTCC DIRECTORY

ACTING DIRECTOR

Rein Mikkor	NP351	5058

INFORMATION OFFICE

General Inquiries	EA206	4990
Marg Doherty	EA206	8995

MANAGER, SERVICES

Stan Yagi	SG200	7331

INFORMATION SERVICES

Don Gibson	EA206	5568

SUPERVISOR, CUSTOMER SERVICES

Bill Foster	SG301	6021

SUPERVISOR, PROGRAMMING SERVICES

Helen Peacock	SG201	6877

SUPERVISOR, APPLICATIONS

Herb Kugel	SG304	7286

SUPERVISOR, CO-ORDINATORS

Ralph Lombardi	SG305	7130

COMPUTING CO-ORDINATORS

Sandford Fleming Terminal		
Mark Tapia	EA207	7109
Program Advisors	EA207	8599
Erindale		
Clem DiPlacido		828-5311
Scarborough		
Ihor Prociuk		284-3173
Arts and Science		
Jill Reece	SS2104	6509
New Physics		
Bob Chambers	NP1201	8823

MANAGER, OPERATIONS

Derry Cox	EA206	7092

SUPERVISORS, OPERATIONS

Paul Scarborough (S/370-165-II)	SF105	6220
Kam Jain (S/360-65)	NP333	7374
Dave Wong (S/370-155)		6864

MANAGER, SYSTEMS

Rein Mikkor	NP351	5058

SYSTEM HARDWARE SUPERVISOR

Eugene Siciunas	EA206	4967

MANAGER, COMPUTER RESEARCH FACILITY

Dennis Smith	SF207	4086

LAB CO-ORDINATOR (CRF)

Krishna Patnaik	SF207	4086

ACCOUNTS

Sylvia May	EA206	7148
Merton Hunte	EA206	8702

ACCESS CODES

Gail Cavers	EA206	8703

MANAGER, INFORMATION SYSTEMS

Sean Keeley		3579

NOTE:

EA = Engineering Annex
SF = Sandford Fleming
SG = 49 St. George
NP = McLennan Physical (New Physics)
SS = Sydney Smith

JOB AND SYSTEM STATUS QUERIES

SYSTEM/370,TSO	7373
ATS/APL	6234
2741 TERMINAL MALFUNCTIONS	7107
029 KEYPUNCH MALFUNCTIONS	6465

SERVICES

	DIAL-UP	ADVICE
APL - 2741	7200	6710
APL(ASCII	7200	6710
ATS	7100	7107
TSO - 2741	7303	8599
TSO - TTY (10 cps)	7390	8599
TSO - TTY (30 cps)	7386	8599
DATATALK	6710	6710
WYLBUR	7235(EBCDIC) 3950(ASCII) CENTREX 3954(ASCII) NON CENTREX	8599

UNIVERSITY OF TORONTO
COMPUTER CENTRE,
TORONTO, CANADA,

CAMPUS DELIVERY

MISS KATTA
SERIALS DEPARTMENT
ROBARTS LIBRARY
UNIVERSITY OF TORONTO